



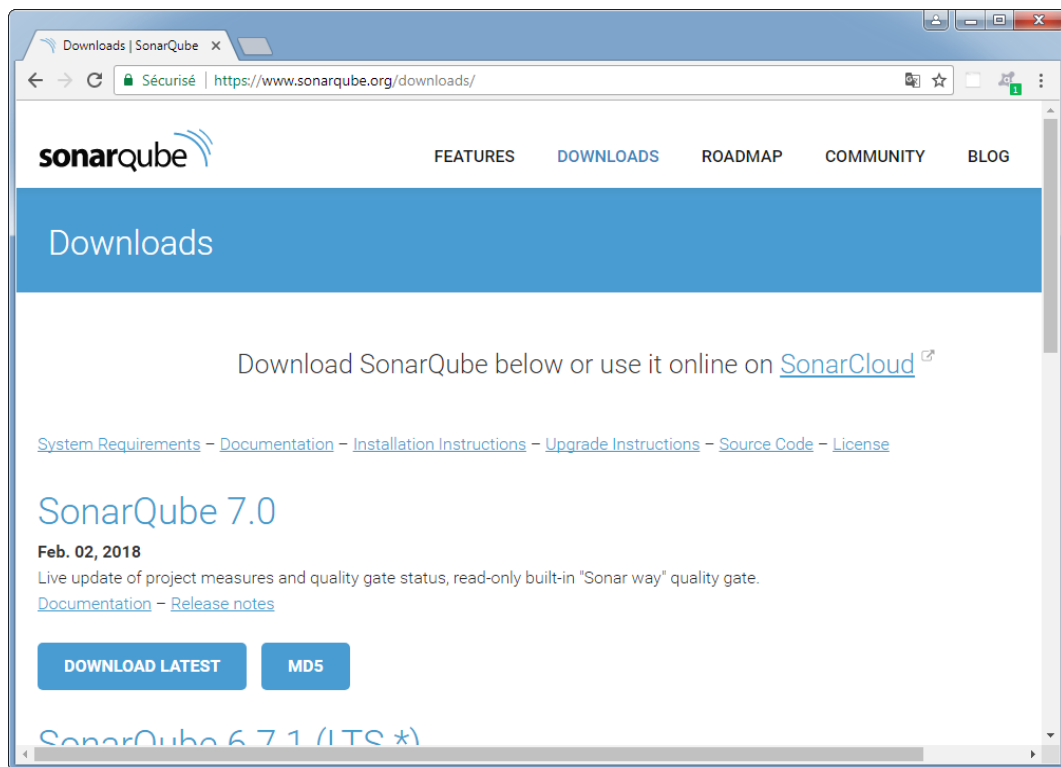
Prérequis

- Java : JDK 1.8 ou plus récent.
- RAM : 256 Mb
- Espace disque : 1 Gb.
- Système d'exploitation : Aucun prérequis.
 - Disponible sous Windows, Linux, MAC, etc...

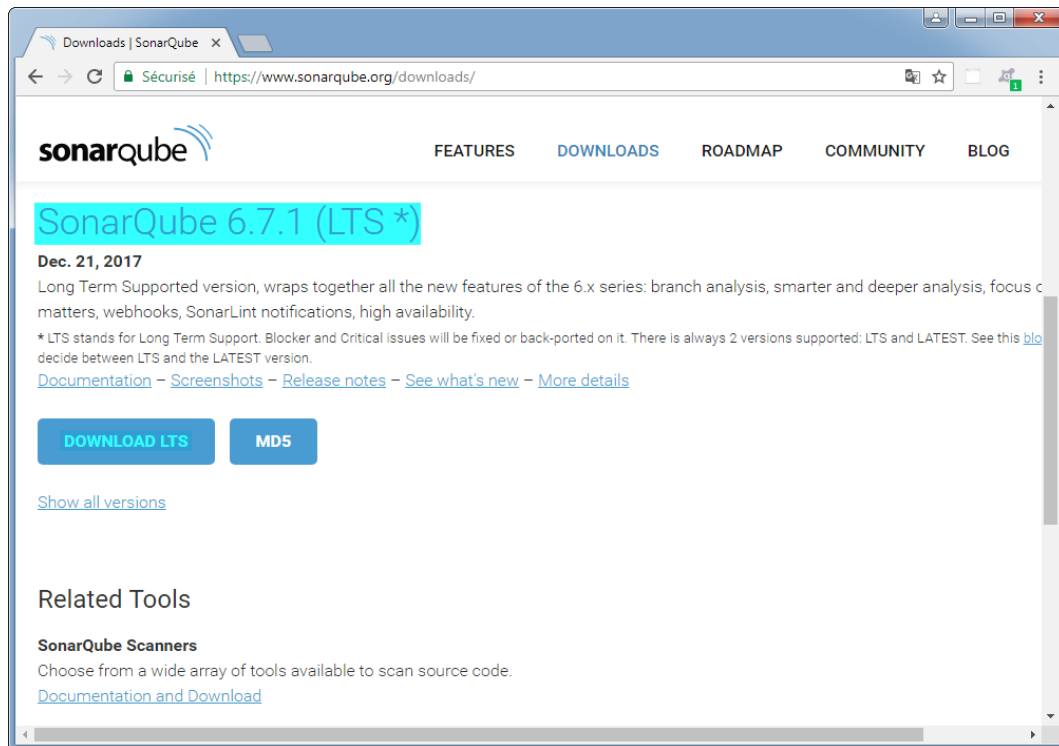
I. Installation du serveur sonarqube

Etape 1: Téléchargement de sonarqube

- Télécharger sonarqube depuis son site web officiel <https://www.sonarqube.org/downloads/>



- Choisir une version Long-term Support (LTS) de type Generic Java package.



Etape 2: Installation de sonarqube

- Télécharger le binaire du sonarqube

```
# wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-5.4.zip
```

- Décompresser et copier le dossier sonarqube-* dans le dossier /opt

```
# unzip sonarqube-5.4.zip
# sudo mv sonarqube-5.4 /opt/sonarqube-5.4
```

- Démarrer le serveur sonarqube

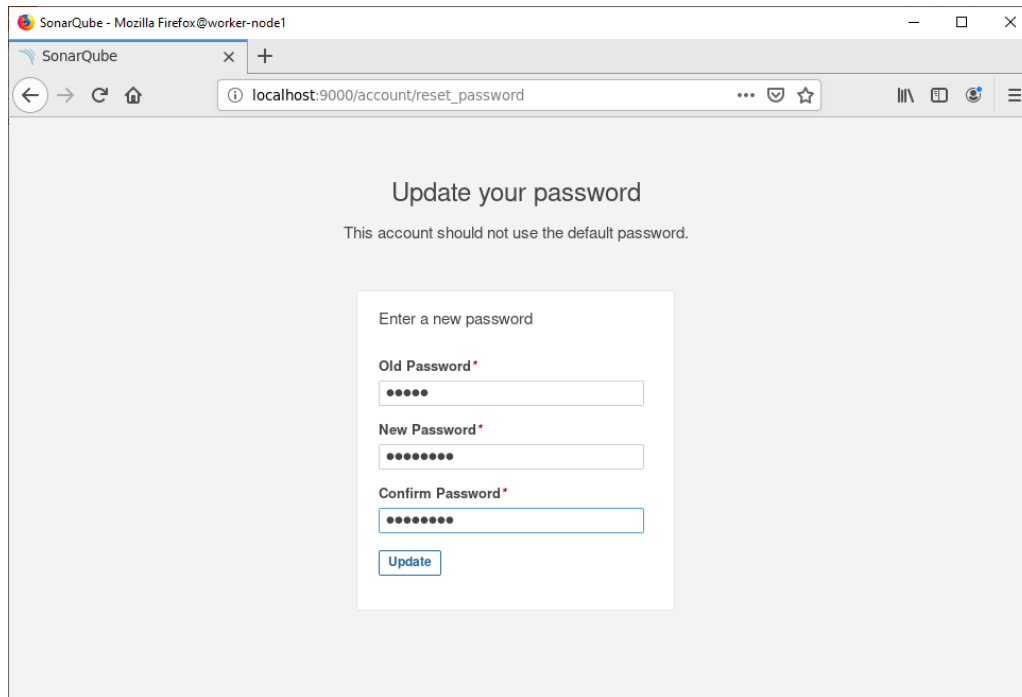
```
# /opt/sonarqube-5.4/bin/linux-x86-64/sonar.sh start
```

```
hassen@hassen-virtual-machine:~$ sudo /opt/sonarqube-5.4/bin/linux-x86-64/sonar.sh start
Starting SonarQube...
Started SonarQube.
hassen@hassen-virtual-machine:~$
```

NB : Pour apporter des modifications sur la configuration globale du serveur sonarqube, éditer le fichier **/opt/sonarqube-8/conf/sonar.properties**

LE SERVEUR D'INTÉGRATION CONTINUE JENKINS

- Testez la connexion au serveur sonarqube via l'url : <http://<IP>:9000>
- Utiliser le compte administrateur par défaut pour s'authentifier : admin/admin puis initialiser le mot de passe



Objectifs :

- Modifier le code source en ajoutant du mauvais code
- Compiler le projet en précisant le goal **sonar:sonar**
- Accéder au tableau de bord du Sonarqube et vérifier les problèmes
- Corriger les erreurs et vérifier le changement de qualité de code sur Sonarqube

Exercice :

- Créer le projet Java « App_Test_Maven_Sonarqube »

```
mvn archetype:generate
```

- Ajouter le plugin sonar et dans le fichier pom.xml

```
<plugin>
  <groupId>org.sonarsource.scanner.maven</groupId>
  <artifactId>sonar-maven-plugin</artifactId>
  <version>3.2</version>
</plugin>
<plugin>
  <groupId>org.jacoco</groupId>
  <artifactId>jacoco-maven-plugin</artifactId>
  <version>0.8.6</version>
</plugin>
```

- Modifier le code source en ajoutant du mauvais code

```
package org.gk.cursusdevops;

import java.util.*;

public class App {

    public App() {
        System.out.println("C'est le constructeur ....");
    }
    public void methodeA() {
        String chaine="";
        List<Integer> list = new ArrayList<Integer>();
        list.add(1);
        list.add(2);
        list=null;
        list.add(3);
        Object obj = getInfo();
        System.out.println(obj.toString());
    }
    public Object getInfo() {
        return null;
    }

    public static void main( String[] args ) {
        System.out.println( "Hello World!" );
        App app = new App();
        app.methodeA();
    }
}
```

- Compiler le projet en précisant le goal :sonar

```
# mvn clean sonar:sonar
```

- Corriger les erreurs et vérifier le changement de qualité de code sur Sonarqube

Page App.java

```
import java.util.*;
public class App{
    public App(){
        //System.out.println("C'est le constructeur ....");
    }
    public void methodeA(){
        //String chaine="";
        List<Integer> list = new ArrayList<Integer>();
        list.add(1);
        list.add(2);
        list.add(3);
        Object obj = getInfo();
        if(obj!=null){
            System.out.println(obj.toString());
        }
    }
    public Object getInfo(){
        return null;
    }
    public static void main( String[] args ) {
        //System.out.println( "Hello World!" );
        App app = new App();
        app.methodeA();
    }
}
```

Page TestApp.java

```
package org.gk.cursusdevops;
import static org.junit.Assert.assertTrue;
import org.junit.Test;

public class AppTest{
    @Test
    public void shouldAnswerWithTrue(){
        App a =new App();
        a.methodeA();
        a.getInfo();
        assertTrue( true );
    }
}
```

Ajouter une fonction methodeB avec un mauvais code

```
public void methodeB(){
    String myString = null;
    System.out.println("Equal? " + myString.equals("foo"));
    return null;
}
```

Correction : `System.out.println("Equal?" + "foo".equals(myString));`

⇒ Cela empêche le déclenchement d'exceptions de pointeur nul, car un littéral de chaîne ne peut jamais être nul par définition.

II. Intégration de Sonarqube avec Jenkins

Etape 1 : Configuration du SonarScanner

- Télécharger le binaire du SonarScanner depuis l'URL : <https://docs.sonarqube.org/latest/analysis/scan/sonarscanner/>

```
# wget https://binaries.sonarsource.com/Distribution/sonar-scanner-  
cli/sonar-scanner-cli-4.5.0.2216-linux.zip
```

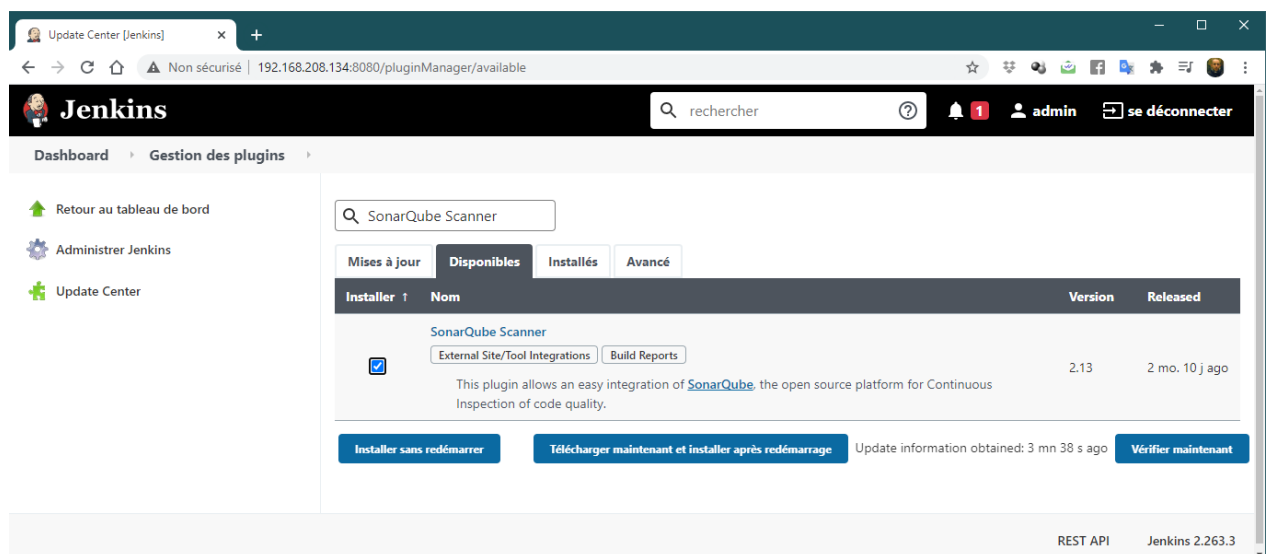
- Décompresser et copier le dossier sonar-scanner-cli* dans le dossier /opt

```
# unzip sonar-scanner-cli-4.5.0.2216-linux.zip  
# mv sonar-scanner-4.5.0.2216-linux/ /opt/sonar-scanner-4.5
```

Etape 2 : Intégration de Sonarqube avec Jenkins

- Installez git plugin (Sans redémarrer)

Administrer Jenkins > Gestion de plugins > Disponibles > SonarQube Scanner



- Configurez de serveur SonarQube:
 - Administrer Jenkins > Configurer le système > SonarQube servers
 - Faites défiler jusqu'à la section de configuration SonarQube, cliquez sur Ajouter SonarQube et ajoutez les valeurs qui vous sont demandées.

LE SERVEUR D'INTÉGRATION CONTINUE JENKINS

Remarque : Le jeton d'authentification du serveur doit être créé en tant qu'informations d'identification 'Secret Text'.

- Cochez la case « Enable injection of SonarQube server configuration as build environment variables »

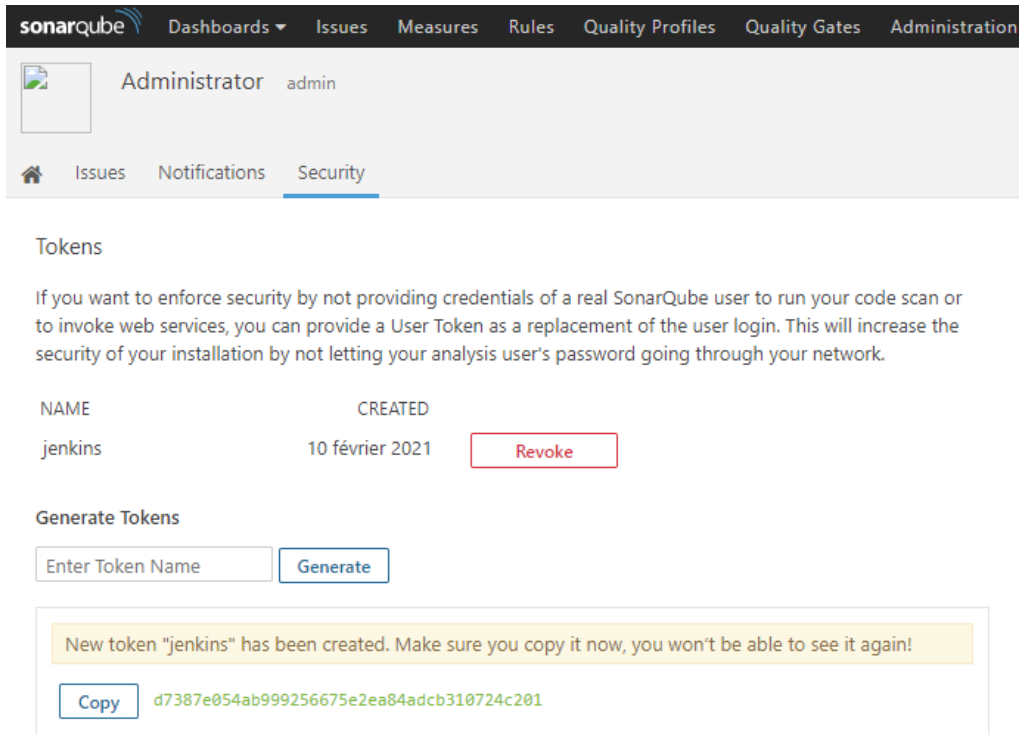
The screenshot shows the Jenkins 'Configure System' page. The 'configuration' tab is active. Under 'SonarQube servers', the checkbox 'Enable injection of SonarQube server configuration as build environment variables' is checked. Below this, there is a table for 'Installations de SonarQube' with columns for 'Nom', 'URL du serveur', and 'Server authentication token'. One installation is listed: 'Sonar5.4' with URL 'http://192.168.208.141:9000/' and token 'sonarToken'. There are buttons for 'Ajouter une installation SonarQube', 'Supprimer installation SonarQube', and 'Avancé...'. At the bottom, there are 'Pipeline Speed/Durability Settings' with a dropdown menu set to 'None - use pipeline default (MAX STABILITY)' and buttons for 'Enregistrer' and 'Appliquer'.

Générer un jeton

Vous pouvez générer de nouveaux jetons dans :

- Utilisateur > Mon compte > Sécurité

LE SERVEUR D'INTÉGRATION CONTINUE JENKINS



The screenshot shows the SonarQube Administration interface. At the top, there's a navigation bar with links: Dashboards, Issues, Measures, Rules, Quality Profiles, Quality Gates, and Administration. Below this, the user is logged in as 'Administrator admin'. The 'Security' tab is selected in the sub-navigation. The main section is titled 'Tokens'. It contains a paragraph explaining that User Tokens can be used instead of real user credentials for code scans. Below this is a table with two columns: 'NAME' and 'CREATED'. There is one entry with the name 'jenkins' and the creation date '10 février 2021'. To the right of this entry is a 'Revoke' button. Below the table is a 'Generate Tokens' section with a text input 'Enter Token Name' and a 'Generate' button. A yellow message box states: 'New token "jenkins" has been created. Make sure you copy it now, you won't be able to see it again!'. Below the message is a 'Copy' button and the token value: 'd7387e054ab999256675e2ea84adcb310724c201'.

NAME	CREATED	
jenkins	10 février 2021	Revoke

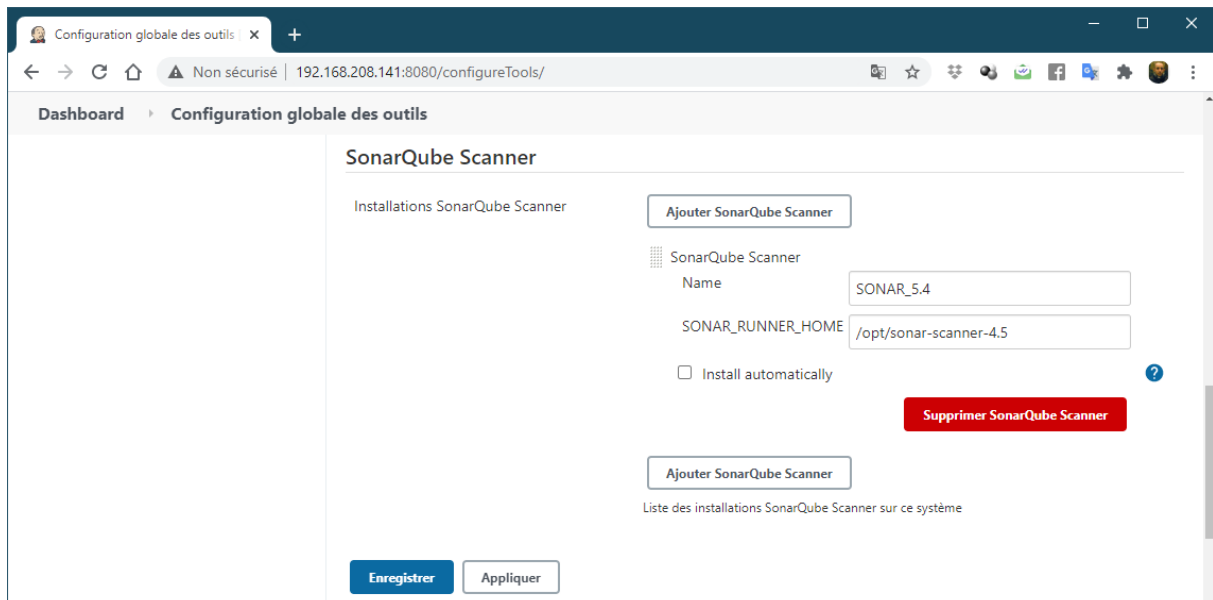
Generate Tokens

Enter Token Name [Generate](#)

New token "jenkins" has been created. Make sure you copy it now, you won't be able to see it again!

[Copy](#) d7387e054ab999256675e2ea84adcb310724c201

- Configurez de SonarQube Scanner:
 - Administrer Jenkins > Configuration globale des outils > SonarQube Scanner



The screenshot shows the Jenkins 'Configuration globale des outils' page for the 'SonarQube Scanner'. The page has a sidebar with 'Dashboard' and 'Configuration globale des outils'. The main content area is titled 'SonarQube Scanner' and shows 'Installations SonarQube Scanner'. There is a table with one entry: 'SonarQube Scanner' with 'Name' 'SONAR_5.4' and 'SONAR_RUNNER_HOME' '/opt/sonar-scanner-4.5'. There is an 'Ajouter SonarQube Scanner' button at the top right and another at the bottom. A red 'Supprimer SonarQube Scanner' button is next to the entry. A checkbox 'Install automatically' is unchecked. At the bottom, there are 'Enregistrer' and 'Appliquer' buttons. The URL in the browser is '192.168.208.141:8080/configureTools/'.

SonarQube Scanner

Installations SonarQube Scanner

[Ajouter SonarQube Scanner](#)

SonarQube Scanner
Name: SONAR_5.4
SONAR_RUNNER_HOME: /opt/sonar-scanner-4.5
<input type="checkbox"/> Install automatically

[Supprimer SonarQube Scanner](#)

[Ajouter SonarQube Scanner](#)

Liste des installations SonarQube Scanner sur ce système

[Enregistrer](#) [Appliquer](#)

- Analyse d'un projet Java
 - Créez un nouvel item de type « projet free-style »

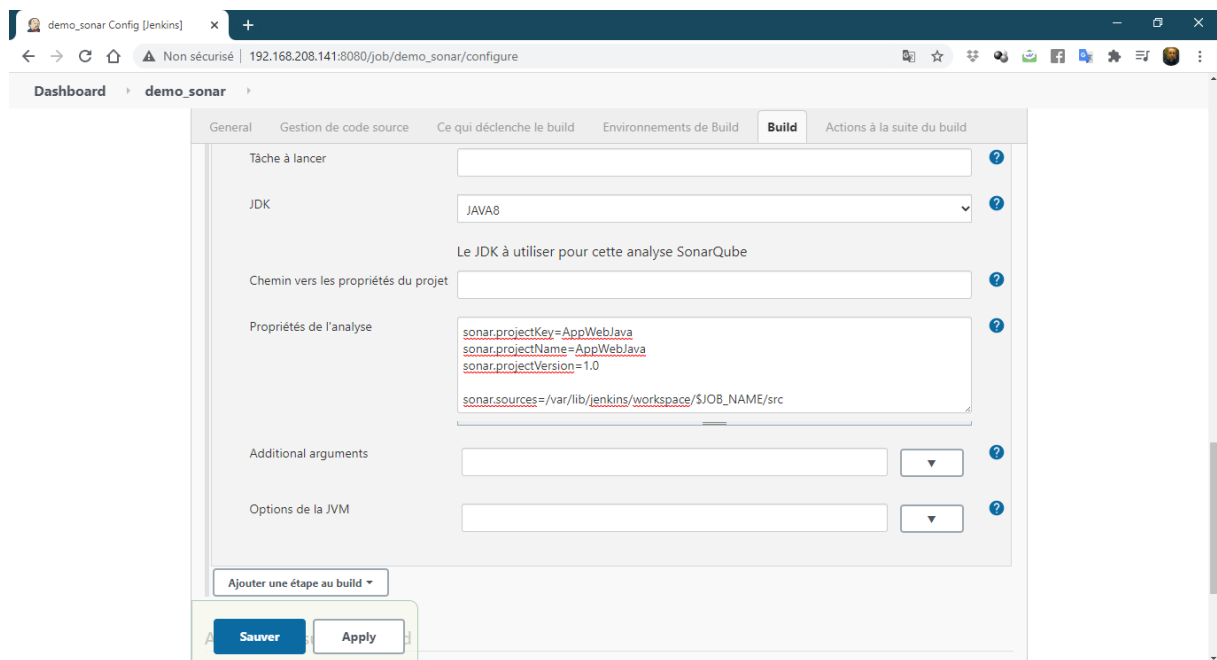
LE SERVEUR D'INTÉGRATION CONTINUE JENKINS

- Dans la section « Build », Ajouter une étape au build de type :
« **Lancer une analyse avec SonarQube Scanner** »
- Remplir les propriétés du projet à analyser :
 - sonar.projectKey
 - sonar.projectName
 - sonar.projectVersion
 - sonar.sources

Exemple :

```
sonar.projectKey=AppWebJava
sonar.projectName=AppWebJava
sonar.projectVersion=1.0

sonar.sources=/var/lib/jenkins/workspace/$JOB_NAME/src
```



- Lancez le build pour démarrer l'analyse de votre projet.

LE SERVEUR D'INTÉGRATION CONTINUE JENKINS

```
INFO: Sensor Code Colorizer Sensor (done) | time=0ms
INFO: Sensor CPD Sensor
INFO: Sensor CPD Sensor (done) | time=0ms
INFO: Analysis report generated in 52ms, dir size=11 KB
INFO: Analysis reports compressed in 31ms, zip size=3 KB
INFO: Analysis report uploaded in 126ms
INFO: ANALYSIS SUCCESSFUL, you can browse http://192.168.208.141:9000/dashboard/index/AppWebJava
INFO: Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
INFO: More about the report processing at http://192.168.208.141:9000/api/ce/task?id=AXeNC68l0TWs26Hsn9lQ
INFO: -----
INFO: EXECUTION SUCCESS
INFO: -----
INFO: Total time: 15.379s
INFO: Final Memory: 9M/71M
INFO: -----
Finished: SUCCESS
```

- Pour vérifier le rapport d'analyse, cliquez sur l'icône SonarQube dans le tableau de bord du projet

